URDUBENCH: A Unified Benchmark for Evaluating Large Language Models on Native Urdu Tasks

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have driven rapid advances in natural language processing (NLP); however, low-resource languages such as Urdu, spoken by over 230 million people, remain severely underrepresented, limiting equitable deployment and widening multilingual performance gaps. Existing Urdu benchmarks are fragmented or translation dependent, lacking a unified framework for evaluating emerging efficient models on native, culturally grounded tasks. We present URDUBENCH, a comprehensive benchmark comprising 20 datasets across 17 tasks for Urdu LLM evaluation, covering natural language understanding, safety-critical moderation, and generation. We also release a modular, open-source evaluation framework enabling reproducible zero-shot evaluation with uniform prompting and metrics. Using this framework, we benchmark 13 open-weight instruction-tuned LLMs spanning nano (<1B), small (1–3B), and medium (up to 7B) parameter scales focusing on models that are computationally efficient and suitable for deployment in low-resource settings. Results show pronounced performance disparities across model sizes and task categories, with persistent difficulties in Urdu sequence labeling and generation, and consistent gains from larger multilingual models.

1 Introduction

Large Language Models (LLMs) have revolutionized natural language processing (NLP), demonstrating remarkable capabilities across a wide array of tasks, from question answering and summarization to reasoning and generation. However, the majority of these advancements have been driven by high-resource languages such as English, where abundant data and benchmarks enable rigorous evaluation and iterative improvements. In contrast, low and mid resource languages, including Urdu spoken by over 230 million people worldwide (SIL International, 2022) as a first or second language remain severely underrepresented in LLM research. This disparity not only limits the equitable deployment of AI technologies but also perpetuates biases and performance gaps in multilingual applications.

At the same time, the ecosystem of open-source LLMs is undergoing a rapid transition toward smaller, efficient models. Sub-billion-parameter models (e.g., gemma-3-270m, granite-4.0-350m, Qwen2.5-0.5B-Instruct) and lightweight 1–3B models (e.g., gemma-3-1b-it, Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct) are increasingly capable of high-quality reasoning and instruction-following, while 4–7B models (e.g., Qwen2.5-7B-Instruct, gemma-3-4b-it) represent the emerging standard for deployable multilingual systems. Despite their practical relevance for resource-constrained settings, there exists *no systematic evaluation suite* designed to measure the Urdu capabilities of these model classes.

To address this gap, we introduce URDUBENCH, a comprehensive benchmark suite comprising 20 datasets across 17 tasks, designed to evaluate LLMs in Urdu. Our benchmark encompasses core NLP competencies, including core structural and classification tasks (e.g., part-of-speech tagging and named entity recognition), ethical and safety-critical tasks (e.g., hate speech and cyberbullying), sentiment and emotion analysis, and generative tasks. Unlike translation-dependent multilingual resources, URDUBENCH integrates datasets that are culturally adapted, manually verified, and natively grounded in Urdu contexts. To facilitate community adoption, we release an open-source codebase for rapid evaluation, featuring a modular framework that allows seamless integration of new models or datasets.

We evaluate URDUBENCH on three model categories: **nano** (under 1B parameters) **small** (1–3B parameters) and **medium** (>3B–7B parameters). Our results reveal significant performance dispari-

ties across model sizes and tasks, underscoring the need for Urdu-specific fine-tuning and highlighting strengths in multilingual base models. Through this work, we aim to accelerate LLM development for Urdu and promote inclusive AI research.

The contributions of this paper are as follows:

- A novel benchmark, URDUBENCH, with 20 datasets spanning 17 tasks tailored for Urdu evaluation.
- An open-source, flexible evaluation framework for rapid testing of LLMs on Urdu tasks.
- Comprehensive analysis of 13 LLMs across three parameter scales, focusing on resource efficient models.

The remainder of the paper is organized as follows: Section 2 reviews related work, Section 3 details the construction of URDUBENCH, Section 4 elaborates the framework of URDUBENCH, Section 5 and Section 6 presents our experiments and results, and Section 7 concludes with future directions.

2 Related Work

Several benchmarks have been proposed to evaluate LLMs across diverse linguistic and cultural contexts. IndicGenBench (Singh et al., 2024) and the Indic LLM Leaderboard (Kumara and CognitiveLab Team, 2024) focus on Indic languages using translated versions of English benchmarks, covering tasks such as summarization, translation, QA, and reasoning. Similarly, SeaExam and SeaBench (Liu et al., 2025) provide native-crafted benchmarks for Southeast Asian languages, targeting both educational assessment and conversational evaluation.

For South Asian contexts, PakBBQ (Hashmat et al., 2025) and PakPolBias (Nadeem et al., 2025) address bias-related evaluation in Urdu and other Pakistani languages, focusing on social and political dimensions. While valuable, these benchmarks are task-specific and do not provide broad coverage of core NLP, safety, and generation tasks for Urdu. (Tahir et al., 2025) evaluated pre-trained LLMs on a range of Urdu NLP tasks; however, the study relied on a static set of now-outdated models and lacked a reusable evaluation framework, making results difficult to reproduce and requiring substantial re-engineering to assess newly released models, thereby underscoring the need for a systematic and extensible benchmarking suite.

Language-agnostic benchmarking frameworks such as LLMeBench (Dalvi et al., 2024) enable rapid evaluation across tasks and languages but do not provide native, culturally grounded datasets for low-resource languages. Broader multilingual efforts include AfroBench (Ojo et al., 2025) and IberBench (Ángel González et al., 2025), which aggregate large collections of datasets across African and Iberian languages, respectively, spanning both general-purpose and region-specific tasks.

Classic English benchmarks such as GLUE (Wang et al., 2019b), WinoGrande (Sakaguchi et al., 2019), and recent dynamic evaluations like LiveBench (White et al., 2024) have shaped LLM evaluation methodology but remain English-centric. Specialized benchmarks such as AD-LLM (Yang et al., 2025) target specific phenomena (e.g., anomaly detection) rather than comprehensive language evaluation.

In contrast to prior work, URDUBENCH provides the first unified, native benchmark for Urdu that jointly evaluates structural NLP, safety-critical moderation, sentiment and emotion analysis, and generation tasks, accompanied by a reproducible evaluation framework tailored to efficient open-weight LLMs.

3 URDUBENCH

URDUBENCH is a unified benchmarking framework designed to support large-scale, reproducible evaluation across a broad range of Urdu NLP tasks and models. The framework is motivated by the need to reduce duplicated engineering effort across experimental configurations while ensuring consistent handling of inputs, intermediate representations, and outputs. URDUBENCH adopts a pipeline-oriented execution model in which information is exchanged through structured key-value mappings, enabling uniform processing across heterogeneous tasks and model providers.

URDUBENCH comprises three core components *Dataset Component*, *Task Component*, and *Model Component* coordinated by a centralized *Benchmark Driver* as illustrated in Figure 1. Execution begins from user-specified configuration, after which the *Benchmark Driver* resolves the selected task and loads associated dataset using the *Dataset Component*. Then get a prompt from the *Task Component* and forwards it to the *Model Component* for

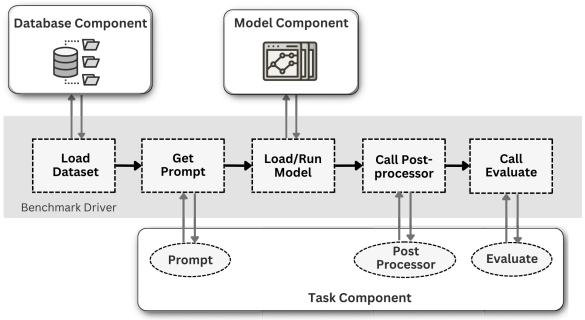

Figure 1: Overall architecture of URDUBENCH. A centralized Benchmark Driver orchestrates Dataset, Task, and Model components under a unified execution pipeline.

inference. Model outputs are returned through a uniform interface, post-processed by the *Task Component*, and evaluated using task-specific metrics. All inputs, intermediate artifacts, and outputs are cached to support reproducibility and efficient re-execution, ensuring consistent evaluation semantics across tasks and models.

To support extensibility while preserving execution consistency, URDUBENCH exposes explicit abstract interfaces for each component. In the following subsections, we describe these components in detail, outlining their responsibilities and interfaces.

3.1 Dataset Component

The *Dataset Component* encapsulates dataset-specific operations, including data loading, preprocessing, and instance structuring. URDUBENCH provides generic loaders for commonly used formats and sources, such as Hugging Face datasets, tabular or structured files such as CSV, TSV, JSON, plain text, and spreadsheet based corpora.

Custom datasets can be incorporated by extending the `DatasetBase` interface (Listing 1). Each dataset implementation requires two things (i) metadata describing dataset provenance, licensing, and language characteristics, (ii) split-aware loading logic that returns a standardized list of evaluation instances. By enforcing a canonical representation of dataset samples, this abstraction enables datasets with heterogeneous formats and storage backends to be evaluated uniformly, without requiring task or model level modifications.

```
class DatasetBase:
    """Base interface for dataset
        loaders."""

    def metadata(self):
        # Returns dataset metadata
        raise NotImplementedError

    def load_data(self, split="test"):
        # Loads and returns a list of
            samples for the specified
            split
        raise NotImplementedError
```

Listing 1: Abstract base class defining the Dataset Loader interface.

3.2 Model Component

The *Model Component* encapsulates model-specific inference operations behind a minimal and uniform interface. URDUBENCH delegates inference configuration, including decoding parameters, prompt handling, and backend-specific execution, to individual model providers. This design supports heterogeneous model architectures and deployment settings while preserving a consistent interaction contract for downstream tasks.

New models are integrated by subclassing the `ModelBase` abstraction (Listing 2), which defines a single mandatory inference method operating over role-based message inputs. To ensure stable large scale benchmarking, the framework provides an optional safety wrapper that prevents runtime failures in individual model calls from interrupting overall evaluation. This lightweight mechanism enables efficient handling of inference errors without coupling failure handling policies to task or dataset logic, making the framework robust to transient issues commonly encountered in practical benchmarking environments.

```
class ModelBase:
    """
    Base abstraction for all model
        providers.
    """

    def prompt(self, messages):
        """
        Perform inference given a list
            of role-based messages.
        """
        raise NotImplementedError

    def safe_prompt(self, messages):
        """
        Optional safety wrapper to
            prevent inference failures
            from interrupting benchmark
            execution.
        """
        try:
            return self.prompt(messages)
        except Exception:
            return "[ERROR]"
```

Listing 2: Abstract base class defining the Model Provider interface.

3.3 Task Component

The *Task Component* represents a reusable evaluation abstraction that defines how inputs are transformed into model prompts, how outputs are interpreted, and how predictions are evaluated. Tasks operate independently of specific model implementations and interact with models exclusively through the standardized model interface.

As illustrated in Listing 3, each task implementation specifies three core operations: prompt construction, output post-processing, and evaluation. URDUBENCH provides default task implementations for common task families such as classification and regression, while allowing custom evaluation logic to be defined when required. Evaluation routines return a dictionary of metric scores (e.g., accuracy, F1, BLEU, ROUGE), enabling flexible reporting and diagnostic analysis. By decoupling task logic from both datasets and models, the framework enables component reuse across diverse benchmarking scenarios with minimal overhead.

```
class TaskBase(ABC):
    """Abstract base for benchmark tasks
        ."""

    @abstractmethod
    def format_prompt(self, sample: dict
        ):
        """Convert a sample into a role-
            based prompt."""
        raise NotImplementedError

    @abstractmethod
    def post_process(self, output: str):
        """Normalize raw model output.
            """
        raise NotImplementedError

    @abstractmethod
    def evaluate(self, refs: list, preds
        : list):
        """Compute evaluation metrics.
            """
        raise NotImplementedError
```

Listing 3: Abstract base class defining the Task interface.

3.4 Execution and Interaction

Once the required components are defined, benchmarking experiments can be executed through a unified command line interface. The framework automatically resolves the appropriate task configuration based on user defined parameters and supports multiple execution modes, including zero-shot evaluation, partial dataset execution for rapid debugging, and cache bypassing for fresh inference. Representative command line interaction patterns are shown in Listing 4.

1. **Single Task-Model Evaluation.** A specific task configuration is executed using a selected model, which supports optional limits on the number of evaluated instances for targeted analysis or debugging.

2. **Model-Centric Evaluation.** A single model is evaluated across all registered tasks, enabling systematic assessment of model robustness and cross-task generalization under a unified execution setup.

3. **Task-Centric Evaluation.** A specific task is executed across all available models, facilitating comparative benchmarking under identical dataset and evaluation conditions.

These interaction modes elevate cross-task and cross-model evaluation to first class execution patterns, enabling scalable experimentation without requiring modifications to task or model definitions.

```
# Single task-model evaluation
python run_bench.py <task_config> --
    model <model_name>

# Model-centric evaluation (model across
     all tasks)
python run_bench.py --model <model_name>

# Task-centric evaluation (task across
    all models)
python run_bench.py --task <task_name>
```

Listing 4: Illustrative command-line interaction modes supported by UrduBench.

4 Framework Design and Properties

URDUBENCH is designed around a set of explicit engineering principles intended to ensure fair, reproducible, and scalable benchmarking in low-resource language settings. The framework enforces uniform execution semantics across all experimental configurations. These design decisions are informed by best practices in modular LLM benchmarking (Dalvi et al., 2024), unified evaluation suites (Gao et al., 2023), and multi-task benchmarks (Wang et al., 2019b,a).

4.1 Modular Asset Design

All experimental components in URDUBENCH, including datasets, task definitions, and model providers, are represented as explicit, self-contained assets that conform to minimal abstract interfaces and are independently registered with the framework. This modular asset design decouples experimental configuration from execution logic, allowing components to be added, replaced, or extended without modifying the core engine, while preventing hidden dependencies between datasets, tasks, and models. As a result, heterogeneous task formulations can be evaluated under identical conditions, ensuring that observed performance differences reflect model behavior rather than ad-hoc evaluation pipelines.

4.2 Unified Task and Model Interfaces

URDUBENCH enforces a strict separation between task logic and model inference through standardized interfaces. *Task Component* define prompt, output post-processing, and evaluation metrics, while *Model Component* provides a minimal inference contract that abstracts away architectural details and deployment backends. This unified interface design ensures that no model-specific assumptions leak into task definitions and that all models are evaluated using identical prompting, decoding, and execution semantics, which is especially critical because minor variations in prompt can substantially affect reported performance (Wang et al., 2019b,a).

4.3 Reproducibility and Robust Execution

To support large-scale experimentation, URDUBENCH incorporates built-in mechanisms for fault-tolerant execution. Model inference is wrapped with configurable caching, enabling experiments to resume seamlessly in the presence of transient failures or infrastructure instability. All experiments are fully parameterized through explicit configuration files and command-line arguments, eliminating implicit state and enabling exact re-execution of experimental setups, while intermediate artifacts and final outputs are persistently stored for transparent reporting and post-hoc analysis.

4.4 Systematic Cross-Product Evaluation

URDUBENCH natively supports systematic evaluation across the Cartesian product of registered tasks and models, providing first-class execution modes for benchmarking a single model across all tasks or multiple models on a fixed task without custom orchestration scripts. By making cross-task and cross-model evaluation a core framework capability, the benchmark discourages selective reporting and simplifies comprehensive analysis of model behavior, while maintaining clear provenance of all evaluated task–model combinations.

4.5 Partial and Debug Execution

To facilitate rapid development and debugging, the framework supports controlled partial execution over subsets of evaluation data. This capability enables verification of task definitions, prompt construction, and metric computation without incurring the computational cost of full-scale inference, while operating within the same execution pipeline as complete runs.

4.6 Language-Independent Design

Although URDUBENCH is motivated by the need for robust Urdu language benchmarking, the framework itself makes no language-specific assumptions. Tokenization strategies, script handling, label schemas, and evaluation criteria are fully delegated to task definitions. This language-independent design allows the framework to be readily extended to other languages with minimal adaptation URDUBENCH as a general-purpose benchmarking infrastructure rather than a language-specific toolchain.

5 Experimental Setup

We conduct a large scale benchmarking study using URDUBENCH to evaluate open weight instruction tuned LLMs across diverse Urdu NLP tasks under zero-shot settings. All experiments were performed on NVIDIA L40 and A100 GPUs using deterministic inference.

5.1 Tasks and Datasets

URDUBENCH is a comprehensive benchmark comprising 20 datasets covering a broad range of Urdu NLP tasks, with a particular emphasis on ethical and safety-critical applications relevant to real-world Urdu online platforms. Figure 2 provides an overview of the benchmark, summarizing the task categories, associated datasets, and the evaluated language models grouped by parameter scale.

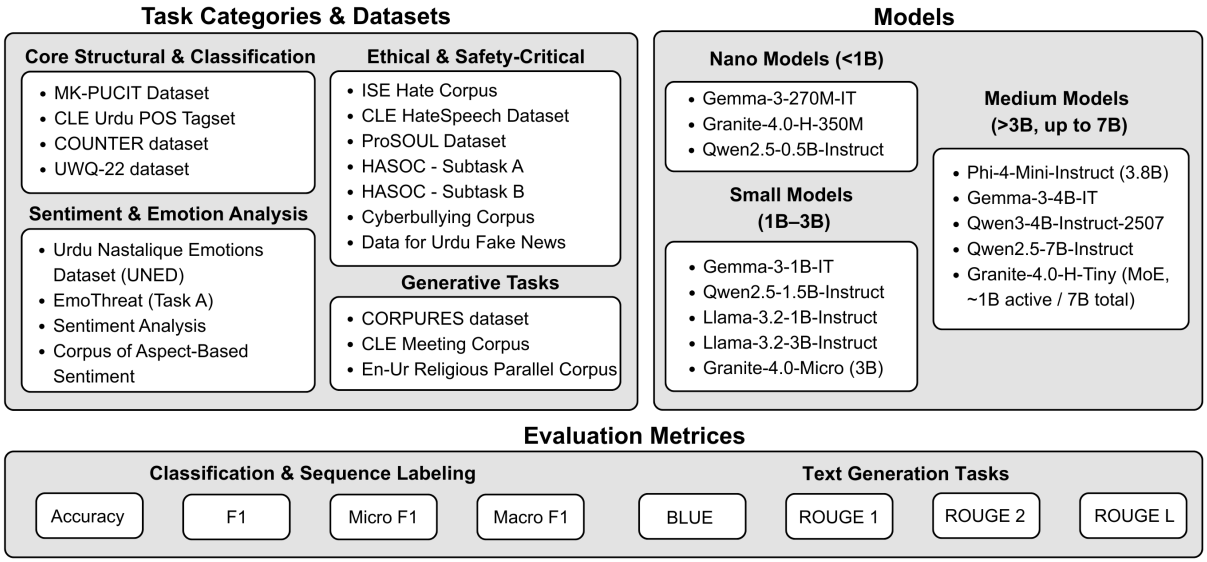

Figure 2: Summary of Task Categories, Datasets, and Model Scales in UrduBench

5.1.1 Core Structural and Classification Tasks

Foundational tasks include named entity recognition using the MK-PUCIT dataset (Kanwal et al., 2019), part-of-speech tagging with the CLE Urdu POS Tagset, news categorization via the COUNTER dataset (Sharjeel et al., 2017), and intent detection using UWQ-22 (Shams and Aslam, 2022).

5.1.2 Ethical and Safety-Critical Tasks

A key focus of URDUBENCH is ethical and safety-critical evaluation. This category covers hate speech detection using the ISE-Hate Corpus (Akram et al., 2023) and CLE-HateSpeech (Ali et al., 2021), propaganda detection with ProSOUL (Kausar et al., 2020), abusive language and threat detection from HASOC subtasks A and B (Das et al., 2021), cyberbullying identification (Adeeba et al., 2024), and fake news detection (Farooq et al., 2023).

5.1.3 Sentiment and Emotion Analysis

Subjective language understanding is evaluated through sentiment analysis datasets supporting both general and aspect-based sentiment (Muhammad and Burney, 2023; ul Haq et al., 2020), alongside emotion classification using the Urdu Nastalique Emotions Dataset (Bashir et al., 2023) and EmoThreat (Task A) (Ashraf et al., 2022).

5.1.4 Generative Tasks

To assess generation in morphologically rich Urdu, URDUBENCH includes extractive summarization using CORPURES (Humayoun and Akhtar, 2022), abstractive summarization via the CLE Meeting Corpus (Sadia et al., 2024), and English–Urdu machine translation based on the English-Urdu Religious Parallel Corpus (Jawaid and Zeman, 2011).

5.2 Model Categories

URDUBENCH evaluates a representative set of open-weight, instruction-tuned large language models spanning multiple parameter scales and architectural families. The benchmark includes models from the Qwen series (Qwen2.5 and Qwen3) (Team, 2024, 2025b), Gemma 3 (Team, 2025a), Granite 4.0 (IBM Research, 2025), Phi-4 (kinfey, 2025), Llama 3.2 (Meta AI, 2024), and the domain-adapted URDULLAMA-1.1-TINY. Models are grouped into three categories based on parameter count: **nano**(<1B), **small**(1B–3B), and **medium** (>3B, up to 7B).

This model selection was done to assess LLM performance under *low-resource and compute-constrained settings*, which are common in real-world deployments. Nano and small models emphasize efficiency, reduced memory footprint, and deployability on limited hardware, making them suitable for edge devices. Medium models are included to study scaling effects while remaining within a practical upper bound for single or few GPU environments. Additionally, the inclusion of URDULLAMA-1.1-TINY, a continued pre-trained model on approximately 800M Urdu tokens built upon Llama 3.2 3B-Instruct, allows direct evaluation of the impact of language-specific adaptation compared to general-purpose multilingual models.

Task	G-270M	Gr-350M	Q-0.5B	G-1B	Q-1.5B	L-1B	L-3B	Gr-3B	L-Tiny	P-4B	G-4B	Q-7B	Q3-4B
Core Classification Tasks													
CLE Urdu Hate Speech	0.438	0.425	0.494	0.456	0.544	0.496	0.533	0.277	0.403	0.556	0.616	0.590	0.585
UWQ-22 Intent	0.293	0.299	0.308	0.320	0.196	0.292	0.329	0.209	0.276	0.377	0.432	0.404	0.328
NER MK-PUCIT	N/A	0.254	0.262	0.337	0.274	0.079	0.198	0.006	0.178	0.414	0.591	0.552	0.586
POS Tagging	0.015	0.037	0.023	0.025	0.101	0.013	0.107	0.005	N/A	0.089	0.131	0.222	0.187
News Categorization	0.106	0.126	0.107	0.552	0.535	0.426	0.771	0.125	0.507	0.719	0.789	0.816	0.801
Extractive Summarization	0.393	0.507	0.480	0.522	0.489	0.312	0.397	0.425	0.368	0.417	0.374	0.313	0.357
Ethical and Safety-Critical Tasks													
CLE Hate Speech Categorization	0.227	0.192	0.214	0.156	0.125	0.092	0.126	0.132	N/A	0.197	0.000	0.338	0.246
ISE Hate	0.378	0.530	0.491	0.382	0.332	0.461	0.534	0.448	0.494	0.670	0.716	0.639	0.716
Abusive Language	0.336	0.507	0.422	0.540	0.337	0.490	0.377	0.374	0.448	0.546	0.624	0.660	0.638
Cyberbullying	0.067	0.120	0.131	0.055	0.079	0.062	0.148	0.000	0.030	0.196	0.242	0.191	0.188
Propaganda Detection	0.542	0.491	0.513	0.472	0.317	0.479	0.628	0.486	0.507	0.551	0.579	0.717	0.618
Threat Detection	0.486	0.506	0.471	0.149	0.258	0.520	0.402	0.246	0.172	0.558	0.455	0.583	0.506
Fake News	0.367	0.370	0.422	0.531	0.517	0.550	0.499	0.542	0.429	0.303	0.671	0.565	0.730
Sentiment and Emotion Analysis													
Sentiment Analysis	0.370	0.245	0.250	0.545	0.367	0.180	0.297	0.258	0.326	0.455	0.635	0.624	0.651
Emotion Analysis Task A	0.293	0.171	0.293	0.277	0.267	0.293	0.242	0.174	0.167	0.342	0.341	0.289	0.339
Paragraph Emotion	0.122	0.065	0.082	0.251	0.299	0.141	0.341	0.167	N/A	0.332	0.158	0.393	0.376
Aspect-Based Sentiment	0.104	0.128	0.170	0.183	0.146	0.064	0.192	0.078	0.170	0.147	0.267	0.286	0.246
Generative Tasks													
Abstractive Summarization	0.032	0.000	0.033	0.000	0.064	0.14	0.034	0.067	0.000	0.000	0.000	0.000	0.150
Bible MT	1.732	0.031	0.065	1.532	0.872	1.532	4.202	5.949	6.501	0.872	0.575	1.910	6.097
Quran MT	0.579	0.000	0.000	0.044	0.258	0.044	0.323	2.710	1.547	0.338	0.000	0.156	2.278

Table 1: URDUBENCH evaluation results across 20 tasks and 13 models. Model abbreviations: G-270M = gemma 3 270M, G-1B = gemma 3 1B, G-4B = gemma 3 4B, Q-0.5B = Qwen 2.5 0.5B, Q-1.5B = Qwen 2.5 1.5B, Q-7B = Qwen 2.5 7B, Q3-4B = Qwen 3 4B, L-1B = Llama 3.2 1B, L-3B = Llama 3.2 3B, P-4B = Phi-4 mini, Gr-350M = granite 4.0 350M, Gr-3B = granite 4.0 micro 3B, Gr-T = granite 4.0 h-tiny 7B.

6 Results and Discussion

We evaluated 13 instruction-tuned LLMs on 20 tasks using URDUBENCH, following the hardware and experimental setup described in Section 5. All evaluations were executed through the unified framework, leveraging fault-tolerant inference, partial-run capability, and reproducible execution semantics. Table 1 reports macro f1-score for core structural and classification tasks, ethical and safety-critical tasks, sentiment and emotion analysis tasks and BLEU score for generative tasks.

6.1 Core Structural and Classification Tasks

In core structural and classification tasks, medium models substantially outperform nano and small models. This pronounced scaling effect is particularly evident in news categorization, where medium models reach up to 0.816 (Qwen-7B), while nano models on average hover around 0.11, suggesting that larger models better capture the lexical and contextual diversity required for accurate Urdu text classification. Named entity recognition and intent classification also follow this trend, with medium models consistently exceeding 0.5 macro-f1, whereas nano and most small models struggle below 0.35 due to insufficient capacity for handling Urdu's rich morphology and script-specific tokenization challenges.

6.2 Ethical and Safety-Critical Tasks

Small models perform relatively well on ethical and safety-critical tasks compared to their weaker results on core structural and classification tasks (nano: 0.381 vs. 0.164; small: 0.354 vs. 0.258), with medium models only slightly better (0.482). Results show that binary signals like hate, abuse, or propaganda transfer effectively from multilingual pretraining even to nano models. Medium models still excel on nuanced subtasks (Qwen2.5-7B-Instruct 0.660 Abusive Language; gemma-3-4b-it 0.716 ISE Hate), but nano models (granite-4.0-350m, Qwen2.5-0.5B-Instruct) often exceed 0.53–0.54 on binary detection, despite near-random performance on core structural and classification tasks. Fine-grained challenges remain across scales (Cyberbullying <0.20; CLE Hate Speech Categorization <0.25), showing socio-cultural subtleties in Urdu toxicity.

6.3 Sentiment and Emotion Analysis

In Sentiment and Emotion Analysis, medium models lead with 0.366 macro-F1 (small: 0.235; nano: 0.191), showing a wider performance gap than in ethical and safety-critical tasks (0.482 vs. 0.366). Small models struggle as these tasks require deeper

contextual understanding of subjective and culturally nuanced language. Medium models excel on sentiment (Qwen3-4B-Instruct-2507 0.651) and paragraph-level emotion (Qwen2.5-7B-Instruct 0.393), while aspect-based sentiment remains challenging across all sizes (<0.27). Emotion classification shows moderate scale gains (Phi-4-mini instruct and medium Qwen/gemma 0.34–0.39), but nano/small models hover near 0.17–0.27, indicating limited transfer from multilingual pretraining.

6.4 Generative Tasks

In Generative Tasks, all models perform poorly, with medium-sized models achieving only marginally higher BLEU scores than smaller counterparts, highlighting a stark contrast with the comprehension oriented tasks elsewhere in the benchmark. While multilingual pretraining enables reasonable understanding and classification of Urdu text even in sub billion parameter models, the ability to produce coherent, contextually appropriate Urdu output remains severely limited across the board. Translation and summarization demand not only linguistic knowledge but also faithful stylistic preservation and cultural nuance, capabilities that current instruction-tuned models lack due to insufficient exposure to high-quality Urdu generative data during training.

7 Conclusion

In this paper, we introduce URDUBENCH, the first comprehensive benchmark suite specifically designed for evaluating large language models on native Urdu tasks. Comprising 20 diverse datasets across 17 tasks including natural language understanding, ethical and safety-critical moderation, sentiment analysis, and generative capabilities. URDUBENCH addresses key linguistic challenges in a low-resource setting without relying on automated translations. We accompany the benchmark with an open-source, modular evaluation framework that ensures reproducible, zero-shot assessments through uniform prompting, post-processing, and metric computation, while facilitating seamless extension to new models and datasets. Our large scale evaluation of 13 open-weight instruction-tuned LLMs across nano (<1B), small (1—3B), and medium (up to 7B) parameter scales reveals stark performance disparities. Larger models, particularly those with strong multilingual pretraining like Qwen and gemma series, consistently outperform smaller ones on classification and generative tasks, though challenges persist in sequence labeling, fine-grained toxicity detection, and long-context generation. Notably, even sub-billion-parameter models show reasonable transfer on binary safety tasks, suggesting opportunities for lightweight moderation in Urdu contexts, while core linguistic and generative competencies demand greater capacity. By releasing URDUBENCH and its evaluation framework publicly, we aim to foster systematic progress in Urdu LLM development, highlight persistent gaps in low-resource multilingual capabilities, and provide a scalable infrastructure extendable to other underrepresented languages. Future work could explore few-shot and propriety models to further chart the path toward equitable AI for Urdu speakers.

Limitations

URDUBENCH is evaluated exclusively in a zero-shot setting. While this enables consistent and reproducible comparison across heterogeneous models, it does not capture performance improvements achievable through few-shot prompting or task-specific fine-tuning on Urdu data. Evaluating such adaptation regimes would provide additional insight into learning dynamics and task sensitivity in future extensions of the benchmark. While the evaluation framework is language agnostic, the current benchmark and experiments are instantiated exclusively on Urdu. The evaluation is further restricted to open-weight lightweight and mid-sized models (up to 7B parameters), reflecting practical deployment considerations in resource constrained environments. Also, the closed-source models are excluded to ensure full transparency and reproducibility. While this design choice enables open comparison, it limits direct assessment against proprietary systems. Incorporating larger-scale and closed-source models, where feasible, remains an important direction for future work.

References

Farah Adeeba, Muhammad Irfan Yousuf, Izza Anwer, Sardar Umair Tariq, Abdullah Ashfaq, and Malik Naqeeb. 2024. Addressing cyberbullying in urdu tweets: a comprehensive dataset and detection system. *PeerJ Comput. Sci.*, 10:e1963.

Muhammad Akram, Khurram Shahzad, and Maryam Bashir. 2023. Ise-hate: A benchmark corpus for inter-faith, sectarian, and ethnic hatred detection on social media in urdu. *Information Processing & Management*, 60.

Muhammad Ali, Ehsan Ul Haq, Sahar Rauf, Kashif Javed, and Sarmad Hussain. 2021. Improving hate speech detection of urdu tweets using sentiment analysis. *IEEE Access*.

Noman Ashraf, lal khan, Sabur Butt, Hsien-Tsung Chang, Grigori Sidorov, and Alexander Gelbukh. 2022. Multi-label emotion classification of urdu tweets. *PeerJ Computer Science*, 8:e896.

Muhammad Farrukh Bashir, Abdul Rehman Javed, Muhammad Umair Arshad, Thippa Reddy Gadekallu, Waseem Shahzad, and Mirza Omer Beg. 2023. Context-aware emotion detection from low-resource urdu language using deep neural network. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(5).

Fahim Dalvi, Maram Hasanain, Sabri Boughorbel, Basel Mousi, Samir Abdaljalil, Nizi Nazar, Ahmed Abdelali, Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Ali, Majd Hawasly, Nadir Durrani, and Firoj Alam. 2024. Llmebench: A flexible framework for accelerating llms benchmarking.

Mithun Das, Somnath Banerjee, and Punyajoy Saha. 2021. Abusive and threatening language detection in urdu using boosting based and BERT based models: A comparative approach. *CoRR*, abs/2111.14830.

Muhammad Shoaib Farooq, Ansar Naseem, Furqan Rustam, and Imran Ashraf. 2023. Fake news detection in urdu language using machine learning. *PeerJ Computer Science*, 9:e1353. Https://doi.org/10.7717/peerj-cs.1353.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2023. A framework for few-shot language model evaluation.

Abdullah Hashmat, Muhammad Arham Mirza, and Agha Ali Raza. 2025. Pakbbq: A culturally adapted bias benchmark for qa.

Muhammad Humayoun and Naheed Akhtar. 2022. Corpures: Benchmark corpus for urdu extractive summaries and experiments using supervised learning. *Intelligent Systems with Applications*, 16:200129.

IBM Research. 2025. Granite 4.0 language models. `https://github.com/ibm-granite/granite-4.0-language-models`. Accessed: 2025-10-01.

Bushra Jawaid and Daniel Zeman. 2011. Word-order issues in english-to-urdu statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95.

Safia Kanwal, Kamran Malik, Khurram Shahzad, Faisal Aslam, and Zubair Nawaz. 2019. Urdu named entity recognition: Corpus generation and deep learning applications.

Soufia Kausar, Bilal Tahir, and Amir Mehmood. 2020. Prosoul: A framework to identify propaganda from online urdu content.

kinfey. 2025. Welcome to the new phi-4 models - microsoft phi-4-mini phi-4-multimodal. Microsoft Tech Community Blog, Accessed: December 31, 2025.

Adithya Sreedharan Kumara and CognitiveLab Team. 2024. Indic LLM leaderboard and IndicEval suite. `https://huggingface.co/spaces/Cognitive-Lab/indic_llm_leaderboard`. Accessed: 2025-12-10.

Chaoqun Liu, Wenxuan Zhang, Jiahao Ying, Mahani Aljunied, Anh Tuan Luu, and Lidong Bing. 2025. SeaExam and SeaBench: Benchmarking LLMs with local multilingual questions in Southeast Asia. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6119–6136, Albuquerque, New Mexico. Association for Computational Linguistics.

Meta AI. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. Accessed: December 31, 2025.

Khalid Bin Muhammad and S. M. Aqil Burney. 2023. Innovations in urdu sentiment analysis using machine and deep learning techniques for two-class classification of symmetric datasets. *Symmetry*, 15(5).

Afrozah Nadeem, Mark Dras, and Usman Naseem. 2025. Framing political bias in multilingual llms across pakistani languages.

Jessica Ojo, Odunayo Ogundepo, Akintunde Oladipo, Kelechi Ogueji, Jimmy Lin, Pontus Stenetorp, and David Ifeoluwa Adelani. 2025. AfroBench: How good are large language models on African languages? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19048–19095, Vienna, Austria. Association for Computational Linguistics.

Bareera Sadia, Farah Adeeba, Sana Shams, and Kashif Javed. 2024. Meeting the challenge: A benchmark corpus for automated urdu meeting summarization. *Information Processing & Management*, 61(4):103734.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Sana Shams and Muhammad Aslam. 2022. Improving user intent detection in urdu web queries with capsule net architectures. *Applied Sciences*, 12:11861.

Muhammad Sharjeel, Rao Nawab, and Paul Rayson. 2017. Counter: corpus of urdu news text reuse. *Language Resources and Evaluation*, 51.

SIL International. 2022. Urdu. https://www.ethnologue.com/language/urd. Ethnologue: Languages of the World (25th edition, accessed 2025).

Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. Indicgenbench: A multilingual benchmark to evaluate generation capabilities of llms on indic languages.

Munief Hassan Tahir, Sana Shams, Layba Fiaz, Farah Adeeba, and Sarmad Hussain. 2025. Benchmarking the performance of pre-trained LLMs across Urdu NLP tasks. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 17–34, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Gemma Team. 2025a. Gemma 3.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025b. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ehsan ul Haq, Sahar Rauf, Sarmad Hussain, and Kashif Javed. 2020. Corpus of aspect-based sentiment for urdu political data.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, and 1 others. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4.

Tiankai Yang, Yi Nian, Li Li, Ruiyao Xu, Yuangang Li, Jiaqi Li, Zhuo Xiao, Xiyang Hu, Ryan A. Rossi, Kaize Ding, Xia Hu, and Yue Zhao. 2025. AD-LLM: Benchmarking large language models for anomaly detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1524–1547, Vienna, Austria. Association for Computational Linguistics.

José Ángel González, Ian Borrego Obrador, Álvaro Romo Herrero, Areg Mikael Sarvazyan, Mara Chinea-Ríos, Angelo Basile, and Marc Franco-Salvador. 2025. Iberbench: Llm evaluation on iberian languages. *Preprint*, arXiv:2504.16921.